## REMARKS

The Examiner has rejected pending Claims 1, 11, 12 and 14 under 35 USC § 103 as unpatentable over the combined teachings of the Boland patent in combination with the teachings of the AAPA. Claims 2, 4-10, 13-16 and 18-20 stand rejected as unpatentable over Boland in view of the AAPA and further in view of the teachings of the Cameron patent. Finally, Claims 3 and 17 have been rejected as unpatentable over Boland in view of the AAPA, Cameron, and Ripps.

The Applicants have addressed all of the cited references before, including in detail in the Preliminary Amendment which was just submitted. Applicants reiterate the arguments below and respectfully request a ***Response to Arguments***, which was not included in the present Office Action. Applicants request the response to provide some guidance as to why the arguments and amendments have not successfully overcome the cited art. Applicants also request, since the present Office Action provided no such guidance, that the next subsequent Office Action not be deemed a Final Office Action.

What is taught and claimed for the present invention is a scheduling method and system for a UNIX-based environment. As is explicitly set forth in the independent claims, at least one local scheduler prioritizes ***processes*** in accordance with a global prioritized schedule which is generated at the global scheduler means. None of the teachings from the cited AAPA art regarding UNIX-based systems provides any teaching or suggestion of such a local prioritizing of processes based on a global schedule.

The present invention provides a system and method wherein a local scheduler maintains a local priority list of ready-to-execute tasks correlated with local processes, which list is updated in accordance with a global prioritized schedule provided from the global scheduler. As encompassed by the independent claims, and as set forth explicitly in the dependent claims, the local prioritized list may be updated by the local scheduler or by the global scheduler. As stated previously, the present approach of global (including inter-node) and local scheduling minimizes unused CPU time when an individual task is temporarily blocked or suspended waiting for I/O. Furthermore, the task used to fill idle time under the present invention is the

next highest task in importance. While some prior art approaches do attempt to interject higher priority tasks for execution at local processors, none of the cited prior art, alone or in combination with each other, provides the apparatus or the method as set forth in the amended claims, including the means and the steps for dynamically creating a global prioritized schedule of said plurality of tasks, said schedule including tasks of said more than one application; communicating said global prioritized schedule to said more than one computing node; determining correspondence between said plurality of tasks and said plurality of local processes; and dynamically prioritizing said local processes in accordance with said global prioritized schedule to allow simultaneous execution of tasks from said more than one application.

The bulk of the teachings of the Boland patent relate to an affinity-based distribution of work in a multiprocessor environment having one node and one scheduler. The single scheduler 22 looks at the global run queue 24 to determine if the next waiting process has affinity with a processor. Based on the determination, the global priority run queue 26 may be updated and a selected

process is sent to the processor. Under those teachings of Boland, it is clear that there is no multinode environment, no local scheduler, no local scheduler prioritized list, no communication of a global prioritized list to local nodes and no updating of a local prioritized list based on the global prioritized list. The Figure 7 embodiment of the Boland patent provides an alternative system wherein the same single scheduler creates both a global run queue for "non-affined" processes and a plurality of nodal run queues for processes having affinity with the node/processor. The processor, as in Zolnowsky, will look at the global queue and look at the local queue and then select the higher priority process. Once again, it is clear that Boland does not create a global prioritized list, does not maintain or update a local prioritized list, and does not actively communicate any priority information (i.e., a global prioritized list) to the local nodes for use in updating a local prioritized list.

While the AAPA is again cited in combination with the primary reference, Applicants maintain the position, as set forth above, that the AAPA references provide the teachings which are missing from the primary reference. Even if one

were to combine the AAPA assignment of same level priorities to all tasks of an application with the Boland teachings, one would not arrive at the present invention since there is no teaching or suggestion of the claim features of a global scheduler for creating and communicating a global prioritized schedule to local schedulers at which local prioritized schedules are updated. Clearly, therefore, Claims 1, 11, 12 and 14 are not obviated by the combination of the cited teachings.

The Examiner goes on to reject Claims 2, 4-10, 13-16 and 18-20 by adding the Cameron patent teachings to the combination of Boland and the AAPA. Applicants rely on the above-stated arguments that the Boland and AAPA references do not obviate the language of the independent claims 1 and 11, from which Claims 2, 4-10, 13-16 and 18 depend. Moreover, the addition of the Cameron patent does not supply the teachings or suggestions which are missing from the earlier-stated combination. The Cameron patent provides parallel tasks but single level global scheduling with no means for deciding what process or task should execute when a single process of the currently-scheduled parallel job is suspended or waiting. Since all actions are initiated from

the Cameron central dispatcher, multiple applications can be assigned to a single processor, however, only one can be active and ready to run at a time. (Applicants reiterate and once again direct the Examiner's attention to the statement in Col. 4, line 5 et seq of Cameron that "although more than one application is assigned in a partition, an entire application is scheduled at once across all the nodes on which it is loaded"). Those applications are assigned by the central dispatcher for the partition. The Cameron global scheduler issues a single directive to execute a task at a particular time and, again, has no capability to prioritize tasks or to dynamically assign tasks of multiple processes in order of importance to utilize idle CPU time. Clearly, therefore, the Cameron patent does not supply the missing teachings to obviate the invention as claimed.

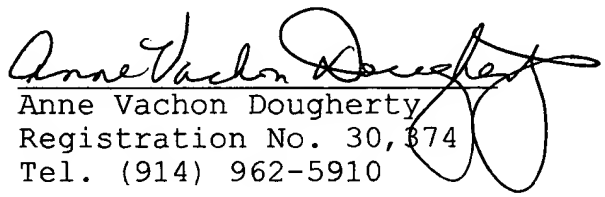
Finally, with respect to Claims 3 and 17, the Examiner has again cited the combined teachings of Boland, AAPA, Cameron, and further cites the Ripps reference. As Applicants had previously argued, the Ripps reference simply provides isolated teachings regarding an operating system functionality. Clearly, the addition of the Ripps teachings does not provide the instruction or suggestion to arrive at

the invention as claimed, since Ripps also does not teach or suggest the use of a global scheduler for creating and communicating a global prioritized schedule to local schedulers at which local prioritized schedules are updated

In light of the foregoing remarks, Applicants respectfully request reconsideration of the arguments, a ***Response to Arguments***, and allowance of the claims.

Respectfully submitted,

M.A. COHEN, ET AL

By: 

Anne Vachon Dougherty
Registration No. 30,374
Tel. (914) 962-5910